ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

A Two Archive Evolutionary Algorithm based on a Vector Angle Approach for Many-Objective Optimization

Maowei He[1, a], Peng Liu[1, b], Liling Sun[2,3, c, *], Donghai Li[3,4, d], and Yabao Hu[5, e]

[1] School of Computer and Technology, Tiangong University, Tianjin 300387, China

[2] Tianjin Innovation and Entrepreneurship Talent Cooperation and Exchange Center, Tianjin 300380, China

[3] Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education, Beijing 100039, China

[4] The Open University of China, Beijing, 100039, China

[5] College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, 300457, China

[a]hemaowei@gotmail.com, [b]liupeng1715@163.com, [c, *]sll198257@gmail.com, [d]lidh@ouchn.edu.cn, [e]huyabao@hotmail.com

Abstract

Multi-objective evolutionary algorithms (MOEAs) confront two challenges when facing with many-objective optimization problems (MaOPs). First, the individuals in population cannot converge to true Pareto optimal front (PF) well. Second, most existing MOEAs have difficulty in keeping population diversity. The two-archive 2 algorithm (Two_Arch2) uses convergence archive (CA) and diversity archive (DA) to balance convergence and diversity for solving MaOPs. Inspired by the idea of Two_Arch2, this paper proposes a two-archive evolutionary algorithm based on a vector angle approach (Two_ArchV) to deal with MaOPs. To address the lack of diversity maintenance in CA, the indicator with boundary protection is used to remove redundant individuals in CA. To simultaneously increase the evolutionary pressure of the DA and improve the convergence and diversity of the final individuals, a vector angle-based approach is applied where the worst-case elimination principle is to replace poorly converging individuals. Two_ArchV is evaluated against various cutting-edge Multi-Objective Evolutionary Algorithms (MOEAs) using WFG and DTLZ benchmarks that have a maximum of 15 objectives. The empirical findings illustrate that Two_ArchV has superior performance compared with other algorithms. In addition, it proves that the incorporation of the two-arcive mechanism enhances the convergence and diversity when solving MaOPs.

Keywords

Multi-objective Evolutionary Algorithms (Maops); A Two-archive Evolutionary Algorithm the Indicator with Boundary Protection; Vector Angle.

1. Introduction

The purpose of a multi-objective optimization problem (MOP) is to achieve the best tradeoff front on many objectives. Many objective optimization problems (MaOPs) [1] are MOPs with more than three objectives. Numerous applications in the real world, including g car controller optimization [2] and automotive engine calibration problems [3]. A multi-objective optimization problem (MOP) can be described as a minimization problem and defined as follows without losing generality [4]:

$$Min\, F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\ \text{subject to } x \in \Omega \tag{1}$$

Where $x = (x_1, x_2, \ldots, x_n)^T$ denotes decision vector in decision space Ω. $F(x)$ is the objective function vector, which is described as $\Omega \rightarrow R^m$. R^m is objective space [5]. When $M > 3$, this problem is known as a MaOPs. Unlike single-objective optimization problems, a MOP is a set of nondominated solutions called a Pareto-optimal set (PS) instead of a single solution. The Pareto optimal front (PF) [6] is formed for $\forall\ x \in PS$ by its objective vector $F(x)$. Multi-objective evolutionary algorithms (MOEAs) [7-9] have been widely and effectively utilized to handle optimization problems with two or three objectives. An essential concept in Multi-Objective Evolutionary Algorithms (MOEAs) is Pareto Dominance. For two different solutions $x, y \in \Omega$. $F(x)$ is said to dominate $F(y)$ (noted as $F(x) < F(y)$), if $\forall m = 1, 2, \ldots, M$, $f_m(x) \le f_m(y)$ and $\exists i = 1,2, \ldots, M$, $f_i(x) < f_i(y)$.

The two-archive algorithm [10] maintains the convergence and diversity by convergence archive (CA)and diversity archive, where each archive focuses on convergence and diversity, respectively. Indeed, the two-archive algorithm was initially proposed to solve the performance decrease of MOEAs when tackling MOPs. This concept is simple, and no extra complexity is added. The nondominated solutions were retained in two archives, CA and DA, in the two-archive algorithm. Specifically, the new nondominated solutions in the whole population with dominance in archives were allocated to CA, whereas other nondominated solutions (without dominance) were assigned to DA. When the total number of members in the archives exceeded the capacity, the removal strategy would be performed, which removed the solution in DA with the shortest Euclidean distance to CA, until the total number met the capacity. However, the number of potential individuals in CA may increase When confronting MaOPs. Since CA and DA sizes are flexible, the members in CA cannot be updated. The update rate of CA is quite low, which might degenerate the overall performance of algorithm. To address the above problems, the improved two-archive algorithm (ITAA) [11] limits CA size. Then, penalty-based boundary intersection (PBI) function assigns fitness values to CA solutions. CA will remove lower-fitness individuals. When DA solutions overflow, shift density estimation-based truncation eliminates them. However, due to flexible DA size, final solutions may have poor diversity performance. To address this issue, Two_Arch2 [12] is suggested as a solution, and CA is updated utilizing the IBEA strategy. When the limit is exceeded by the individuals in CA. DA adopts L_p-norm-based similarity method to delete extra solutions. Finally, DA is produced as TwoArch2's final output. Cai [13] adopts two different updated techniques based on the aggregation-based framework. The diversity of DA may have poor distribution for with partial PFs. Since various weight vectors may result in the same optimal solution.

Although some results have been achieved in this field, the study of MaOPs is still needs to be explored. Convergence and diversity are two important but contradicting objectives. How to design a strategy that strikes a balance between convergence and diversity is the core of an efficient MaOEA. In Two_Arch2, as the number of objectives increases, the efficiency of Two_Arch2 based on Pareto dominance drops sharply. The individuals in CA are updated by the quality indicator of IBEA, which can improve the convergence but lacks diversity maintenance. Moreover, for the archive DA, L_p-norm-based distance strategy cannot ensure a good distribution. The convergence and diversity of individuals in DA has poor performance when handling MaOPs.

To address the above problems, a vector angle-based two-archive algorithm (Two_ArchV) is proposed in this paper, Two_ArchV follows the core framework of Two_Arch2. First, the CA and DA are randomly initialized. Crossover is executed between CA and DA, but mutation is only performed in DA during the reproduction process. However, different strategies are used in CA and DA updating process. In CA updating process, the boundary protection strategy based on $I_{\varepsilon+}$ indicator is adopted to remove reductant individuals, which could improve diversity of individuals without undermining population convergence. In DA updating process, a vector angle-based approach becomes a measure of individual performance. Where individuals with poor convergence (measured by the sum of normalized objectives) are replaced by other good individuals in term of convergence.

The following is a summary of the main contributions of this paper.

1) $I_{\varepsilon+}$ indicator based a boundary protection strategy is used to CA truncation, which improves diversity of CA.

2) Unlike decomposition-based algorithm, Two_ArchV does not use weight vectors or reference points. A vector angle-based strategy is applied in updating DA, which consists of two parts: maximum-vector-angle-first principle for ensuring the good distribution of the population by means of vector angles and the worse-elimination principle for making the poorly converged solutions to be replaced by those good solutions one by one.

The remainder of this paper is organized as follows. Section 2 presents some basic flow of Two_Arch2, and the strengths and drawbacks are analyzed for improvement. Section 3 introduces some details about the proposed algorithm. Section 4 provides test problems and settings of test parameters. Section 5 shows discussions and analysis of the experimental results. Section 6 is a summary of proposed algorithm.

2. Algorithm of Two_Arch2

The two-archive algorithm [10] presents a framework for dividing the non-dominated solution set into two distinct archives, convergence archive (CA) and diversity archive (DA), which focus on convergence and diversity, respectively. After obtaining new solutions by reproduction operation (crossover and variation), CA and DA are updated by non-dominated offspring solutions. Non-dominated offspring solutions with dominance are added to CA, while non-dominated solutions without domination are added to DA. The total number of CA and DA is flexible. When the total size of CA and DA overflows, the solutions in DA with the minimum distance to CA are deleted. Inspired by Two Arch, the Two_Arch2's CA and DA, which are are no longer flexible [12]. CA and DA have a fixed size. As a quality indicator of IBEA [14], it guides the population to PF quickly in CA. DA selects non-dominated solutions according to Pareto dominance. In order to increase diversity, the *Lp*-norm-based distances are applied after DA exceeds.

2.1 Basic Flow

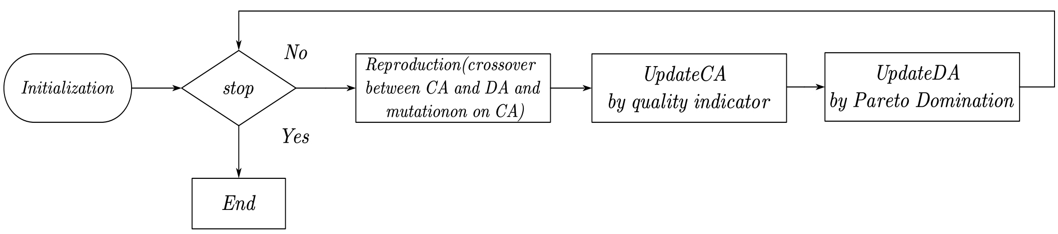

Fig. 1 Flowchart of Two_Arch2 is shown.

Fig. 1 depicts the flowchart of Two_Arch2. CA and DA are first initialized at random before generating offspring by simulated binary crossover (SBX) and polynomial mutation (PM) [15]. Unlike conventional MOEAs, the Two_Arch2 algorithm only performs crossovers between CA and DA, whereas mutation is conducted on CA. DA and CA have a fixed size. The CA's goal is to direct the population into convergent PF, therefore the CA is updated by quality indicator. While the DA intends to keep diversity in a high-dimensional objective space by a new L_p-norm distance strategy. When DA overflows, the non-dominated solutions are first retained in the DA, and then the boundary solution with the maximum and minimum objective values is added. Finally, the solutions that is most different from the selected solution is added to the DA according to the L_p -norm distance. Two_Arch2 algorithm use DA as the final output.

2.2 Advantages and Detriments

The Two_Arch2 algorithm assigns the roles of convergence and diversity to CA and DA, respectively. Two archives are assigned distinct selection principles. In updating CA, the quality indicator of IBEA is utilized. In updating DA, L_p-norm-based distance is adopted to truncate the redundant individuals. The core idea is very simple and does not require additional parameter settings.

Two_Arch2 has excellent performance in terms of convergence and diversity of the final solutions in the Pareto front. Nevertheless, the effectiveness of Two_Arch2 diminishes as the number of targets increases due to Pareto dominance degeneration. For the archive CA without any diversity maintenance mechanism, overemphasizing convergence would cause that CA is unable to sustain diverse individuals. For the archive DA, the individual selection based on Pareto dominance may not offer sufficient selection pressure in high-dimensional objective space. Meanwhile, L_p-norm-based distance is adopted to maintain a good diversity, which may make the individuals of DA offer the poor distribution of potential individuals when handling MaOPs.

3. Proposed Algorithm

This section introduces a modified technique called the Two_ArchV, which is based on a vector angle method. The purpose of this algorithm is to solve many-objective optimization problems (MaOPs). In Two_ArchV, the framework of Two_Arch2 is utilized. In order to balance the diversity and convergence of CA and DA when solving MaOPs, a $I_{\varepsilon+}$ indicator based on the boundary protection strategy is used to enhance diversity of CA and a vector angle-based approach is employed in balancing convergence and diversity of DA.

3.1 Main Structure of Algorithm

The framework of Two_ArchV is similar to that of Two_Arch2, as shown in Fig. 2. Firstly, the CA and DA are randomly initialized. In the process of reproduction [15], Two_ArchV is able to make crossovers between CA and DA, although mutations on CA are only made during this process. Different from the Two_Arch2, there are some significant improvements in the proposed Two_ArchV.

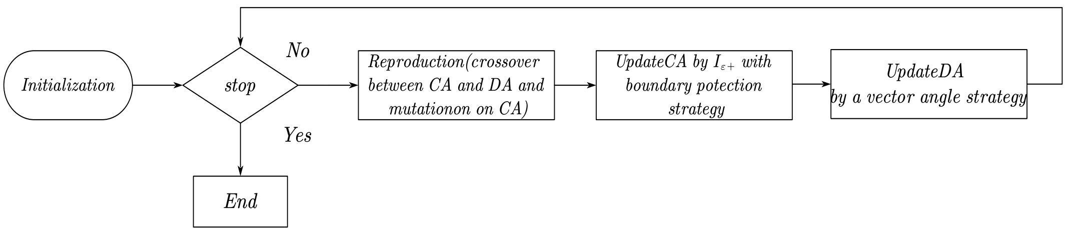

Fig. 2 Flowchart of Two_ArchV is shown.

The a new $I_{\varepsilon+}$ indicator with boundary protection is assigned in updating CA, that could improve diversity of individuals without undermining population convergence. In this way, there are two selection procedures used to update individuals in CA: the $I_{\varepsilon+}$ indicator value for each two individuals and the L_p-norm distance value from individual to presupposed curve/surface.

A vector angle-based approach is employed in DA. The nondominated sorting method divides all the individuals in DA into different levels. Once the last level, F_l, is identified, the principle of vector angle maximum is used to select individuals with maximum vector angle from F_l and put them into DA one by one. This method first ensures a good distribution of individuals in the DA. To enhance the selection pressure of DA individuals in high-dimensional space, some individuals with the same search direction are replaced by good individuals in term of convergence. It is known as the worst-elimination principle.

Algorithm 1: Two_ArchV

Input: a MaOP; archive size N
Output: DA
1: Randomly generating initial archives CA and DA
2: $t \leftarrow 0$
3: **while** $t < \text{Gen}_{max}$ **do**
4: Produce a population of offspring O By means of reproduction
5: $CA \leftarrow \text{updateCA}(CA \cup O)$
6: $DA \leftarrow \text{updateDA}(DA \cup O)$
7: $t \leftarrow t + 1$
8: **end while**
9: **return** DA

The framework of the Two_ArchV algorithm is presented by Algorithm 1. First of all, N solutions are randomly generated as initial archives of CA and DA (line 1). In each generation, CA and DA as parents generate offspring by simulated binary crossover and polynomial variation. Ultimately, CA and DA archives are updated based on different method, until the iteration condition is met (lines 5 and 6). DA is regarded as the final output.

3.2 Convergence Archive

Because the CA in Two_Arch2 lacks a mechanism for maintaining diversity, a new indicator with the boundary protection in the MOEA-IBP algorithm [16] is chosen as the CA selection strategy in Two_ArchV. In this paper, $I_{\varepsilon+}$ indicator with boundary protection effectively addresses the problem of poor diversity in CA. The pseudocode of the updating procedure is shown in Algorithm 2.

Algorithm 2: update CA procedure

Input: CA, O (offspring set), N (CA size)
Output: CA_{upd}
$CA \leftarrow CA \cup O$
T = ParetoNondomination Levels (CA)
Estimate the L_p-norm distance, denoted as $Dis(x, C)$ between each member in T and the surface C
while $|T| > N$
Estimate $I_{\varepsilon+}(x_a, x_b)$ for each pair of individuals using (1), and then identify the pair with the lowest $I_{\varepsilon+}$ value in T.
if $I_{\varepsilon+}(x_a, x_b) < 0$
Delete x_b from T
else if $I_{\varepsilon+}(x_b, x_a) < 0$
Delete x_a from T
else
if $Dis(x_a, C) > Dis(x_b, C)$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

```
12:             Delete x_a from T
13:         else if Dis(x_a, C)< Dis(x_b, C)
14:             Delete x_b from T
15:         else
16:             Randomly Delete the x_a or x_b
17:         end if
18:     end if
19:   CA_upd ← T
20: end while
21: return CA_upd
```

First, in order to select CA for the next generation, CA and the offspring set *O* are combined (line1). Second, the update strategy is carried out. After that, the individuals in the CA are sorted using fast non-dominated sorting approach and divided into different levels (F_1, F_2, F_3 and so on). All individuals in level 1 to level *l* are added to the temporary population *T* (line 2) until the size of *T* exceeds *N* for the first. Finally, $I_{\varepsilon+}$ quality indicator is the first selection principle to select elite individual. The $I_{\varepsilon+}$ (x_a, x_b) values of each two individuals are calculated as follow:

$$I_{\varepsilon+}(x_a, x_b) = arg\ min_{\varepsilon}\left(f_i(x_a) - \varepsilon \leqslant f_i(x_b)\right),\ for\ i \in \{1,\cdots,m\} \tag{2}$$

Where the pair of individuals x_a, x_b with the minimum value of $I_{\varepsilon+}$ will be selected in *T*. If $I_{\varepsilon+}$ (x_a, x_b) < 0 or $I_{\varepsilon+}$ (x_a, x_b) > 0 means $x_a < x_b$ or $x_a > x_b$, the dominated individuals will be removed from *T* (line 6-9). Otherwise, an implementation of the border protection approach, which serves as the second selection principle.

The border protection approach is inspired by KnEA algorithm [17]. This technique involves calculating the distance *Dis*(*x*, *C*) between the individuals in *T* and a surface *C* to assess the fitness of the individuals. The surface *C* is stated as follows:

$$\left(\sum_{i=1}^{m}\left(f_i(x)\right)^P\right)^{\frac{1}{p}} = 1 \tag{3}$$

Let $f_i(x)$ repersent a vector on the surface *C*, and let *P* be the the curvature of the *C*. Furthermore, L_p-norm distance is used as an approximation for the *Dis*(*x*, *C*).The L_p-norm distance between the individuals on surface *C* and the origin point is equal to 1. The calculation of the approximation of *Dis* (*x*, *C*) is calculated as follow:

$$d(x,C) \approx \left(\sum_{i=1}^{m}\left(f_i(x)\right)^P\right)^{\frac{1}{p}} - 1 \tag{4}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

A lower value of $Dis(x, C)$ signifies a shorter distance between the individuals and the surface. Additionally, $Dis(x, C)$ takes on a negative value for individuals positioned below C. When updating CA, if the $I_{\varepsilon+}$ indicator is unable to distinguish the superior individual, the individual with the higher value of $Dis(x, C)$ is eliminated from the T. If neither of them can show the difference between the two individuals, one of them is randomly taken out of T (line 10-17). The updating process is repeated until the CA reaches N.

3.3 Diversity Archive

In the Two_Arch2, DA's updating strategy is built on Pareto dominance, which means that there is less selection pressure on a high-dimensional objective space. The diversity-based selection, specifically using *Lp*-norm-based distances in Two_Arch2, will have a significant impact on determining which individuals survive.As a result, the final population might have a good distribution, but it might also be quite a distance from the PF that was anticipated. To alleviate the above phenomenon, DA is assigned as the selection strategy with the vector angle-based approach. In this approach, individuals with poor convergence are replaced by other individuals with good convergence.

The general structure for the update of the DA method is presented in the pseudo-code of Algorithm 3. To begin with, DA and the offspring set O are combined to select the elite individuals for next generation. DA is normalized as shown in (5) (line 3 in Algorithm 3).

$$f_i'(x_j) = \frac{f_i(x_j) - z_i^{\min}}{z_i^{\max} - z_i^{\min}} \tag{5}$$

Where m is the number of objectives and the ideal point $Z^{\min} = (z_1^{\min}, z_2^{\min}, \ldots, z_m^{\max})^T$ is the minimal value of each objective for every individual in DA. Similarly, the nadir point $Z^{\max} = (z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max})^T$ can be calculated. For each individual $x_j \in$DA (j = 1,2,3, …,2N), its objective vector is $F(x_j) = (f_1'(x_j), f_2'(x_j), \ldots, f'_{\mathrm{m}}(x_j))^T$.

After the normalization of DA, the fitness value of each individual in DA ($x_j \in$DA) is estimated. This fitness value takes into account the total of the normalized objective values through following equation [18].

$$fit(x_j) = \sum_{i=1}^{\mathrm{m}} f_i'(x_j) \tag{6}$$

This fitness value provides an approximate measure of the convergence information. The estimation function is controlled by two key factors: the quantity of objectives and the performance achieved in each individual objective. The performance of an individual improves as the fitness value decreases.

Next, DA is divided into different levels ($F_1, F_2, F_3, \ldots, F_l$) by fast nondominated sorting approach. All individuals from the level 1 to the level l are first added to the temporary set S_t. Therefore, the final front F_l is identified (lines 4-6). If the size of S_t is exactly N, the updated DA is returned. Otherwise, K individuals in the F_l are sequentially added into P utilizing the vector-angle-maximum selection and worst-termination approach of function (lines 11-14).The following provides a detailed description of DA update procedure main components.

Algorithm 3: update DA procedure

Input: *DA, O* (offspring set), *N* (DA size)

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

Output: DA_{upd}

$DA \leftarrow DA \cup O$

$S_t = \varnothing$,$i = 1$ // Generate a temporary population S_t

Normalization of DA

Pareto Nondomination levels(DA)

$| S_t = S_t \cup F_i |> N$ and $i = i+1$

The last front $F_l = F_i$

If $|S_t| = N$

$updated\_DA \leftarrow S_t$

return *updated_DA*

Else

Select $K = N - |S_t|$ individuals from F_l and add them added to S_t

Establish a correlation between each member of F_l and the corresponding individual in S_t.

S_t = vector-angle-maximum selection (S_t, K, F_l)

S_t = worst-truncation strategy (S_t, K, F_l)

$DA_{upd} \leftarrow S_t$

End If

Return DA_{upd}

Definition (Vector Angles): After normalizing the objective vector of DA, the vector angles between two solutions in the normalized objective is defined as:

$$\text{angle}(x_c, y_d) \equiv \arccos\left|\frac{F'(x_c)\cdot F'(y_d)}{\|F'(x_c)\|\times\|F'(y_d)\|}\right| \tag{7}$$

where $F'(x_c)\cdot F'(y_d)$ presents the inner product of two vector, and $\| \|$denotes the norm of a vector.

Definition (Extreme solution): The solution in the objective space that has the minimum angle to the coordinate axis is called the extreme solution. The extreme solution for the r-th axis is defined as the one with the minimum vector angle to the axis vector $v = (v_1, v_2, \ldots, v_m)^T$.

$$e_r = \text{argmin}\{\text{angle}(x, v)\} \tag{8}$$

$$v_i = \begin{cases} 1, & i = r \\ 1e-6, & i \neq r \end{cases} \quad i \in \{1, 2, \ldots, m\} \tag{9}$$

The Association operation is able to connect the members of F_l with the members of S_t according to the vector angles. We find all minimum angles between each member in final front F_l to individuals of population S_t. The minimum angle denotes as $\theta(x_c)$, where $x_c \in F_l$. Then, the solution in S_t to which x_c has the smallest vector angle is known as the target solution of x_c, and the index of target solution

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

is $\beta(x_c)$. It is worth noting that as the number of objective increases, the nondominated sorting fails to classify individuals in the population into different levels. Thus all individuals fall into the last front F_l. In that case, we first remove m extreme solutions from F_l and add them to S_t. After that, we take into account and add to S_t the first m converged solutions that have the best fitness value according equal (6).

1) The vector-angle-maximum selection is used to select candidates with maximum vector angle from F_l one by one to fill the S_t. Algorithm 4 presents the process of this selection. Specifically, Each individual in F_l has a flag value that shows whether they have been added to S_t or not.At the beginning of the procedure, all individuals in the F_l are initialized as 0. The solution of F_l that has the maximum vector angles to S_t is found, and the index of this solution is denoted ρ. Afterward, we add the best individual x_ρ with maximum vector angles to S_t and set its flag to ture (lines 3-7 in Algorithm 4). It is necessary to update the vector angles between the new S_t and the existing solutions in F_l. To achieve updating of process, the vector angle between x_ρ and every solution in F_l with flag value 0 needs to be identified. If the vector angle is smaller than original one, the update process is executed (lines 11-16 in Algorithm 4). This selection method chooses individuals dynamically and is supposed to maintain a well-balanced population.

Algorithm 4: P = The vector-angle-maximum selection (S_t, K, F_l)

Input: S_t, K, F_l
Output: The New S_t
1: $k = 1$
2: **While** $k \le K$ **do**
3: $T = |F_l|$
4: $flag(x_c) == 0$ where $x_c \in F_l$, $c = 1, 2, \ldots, T$
5: $\rho = max\{\theta(x_c) | x_c \in F_l \wedge (flag(x_c) == 0)\}$ //index with the maximum vector angle
6: $S_{t}, = S_{t}, \cup x_\rho$
7: $flag(x_\rho) = true$
8: **For** $j=1$ *to* T
9: **If** $flag(x_c) == 0$
10: The *angle* is calculated (*angle* between x_c and x_ρ)
11: **If** $angle < \theta(x_c)$
12: $\theta(x_c) = angle$
13: $\beta(x_c) = | S_{t},|$
14: **End If**
15: **End If**
16: **End for**
17: $k = k + 1$
18: **End While**

2) The worst-truncation strategy involves selectively replacing individuals with poor convergence in order to enhance the convergence of the DA. The pseudo-code of the worst-truncation strategy is shown in Algorithm 5. In order to evaluate an individual's convergence performance, the total of all normalized objectives is taken into consideration according to Equation (6). When an individual in P is replaced by a member in F_l, the remaining target solutions are also updated. When two solutions

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

have an angle that is less than thethreshold $\sigma = ((\pi/2)/N + 1)$ [19], they are considered to be searching in the same direction.

To be specific, each individual in the last level F_l exists a flag value that indicates whether this individual has been added to S_t or not There is a flag value for each solution in F_l that shows whether this solution has been added to S_t or not (line 4 in algorithm 5). We find the individual in F_l that has the smallest vector angles to S_t. The symbol μ is used to represent this individual's index.Then, the target solution $y_d \in S_t$ is replaced x_μ, when $angle(x_\mu, y_d) < \sigma$ and $fit(x_\mu) < fit(y_d)$ in Equation. (6). After replacement, x_μ in S_t may be selected in the next generation. There are two scenarios to be considered here:

a) When $\beta(x_c) \neq \beta(x_\mu)$, it indicates that x_c and x_μ are connected to distinct solutions in S_t. In such a case, if the *angle* $(x_c, x_\mu) < \theta(x_c)$, then, $\theta(x_c)$ and $\beta(x_c)$ are updated, respectively. Otherwise, they remain unchanged (lines 13-17 in algorithm 5).

b) When $\beta(x_c) = \beta(x_\mu)$, it represents that the both x_c and x_μ are connected to the same member in S_t. In this instance, the value of $\theta(x_c)$ needs to be modified (line 19).

Algorithm 5: S_t = Worst-truncation strategy (S_t, K, F_l)

Input: S_t, K, F_l
Output: The New S_t

1: $k = 1$
2: **While** $k \leq K$ **do**
3: $T = |F_l|$
4: $flag(x_c)$ == false where $x_c \in F_l$, $c = 1, 2, \ldots, T$
5: $\mu = min\{\theta(x_c) | x_c \in F_l \wedge (flag(x_c) == 0)\}$ //index with the minimum vector angle
6: $d = \beta(x_\mu)$ // y_d is the dth individual in S_t
7: **If** $(\theta(x_\mu) < \sigma) \wedge (\text{fit}(x_\mu) < \text{fit}(y_d))$
8: Replace y_d with x_μ
9: $flag(x_\mu)$ = true
10: **For** $j = 1$ to T
11: **If** $flag(x_j) == 0$
12: angle between x_c and x_μ is calculated
13: **If** $\beta(x_c) \neq \beta(x_\mu)$
14: **If** $angle < \theta(x_c)$
15: $\theta(x_c) = angle$
16: $\beta(x_c) = d$
17: **End If**
18: **Else**
19: $\theta(x_c) = angle$
20: **End If**
21: **End If**
22: **End for**
23: **End If**
24: **End while**

4. Experimental Design

4.1 Problems and Performance Metrics

In order to verify the performance of the proposed Two_ArchV, DTLZ [20] benchmark suites and WFG [21] benchmark suites are employed in this experiment. DTLZ1 to DTLZ4 and WFG1 to WFG9 with 3,5,8,10,15 objectives are chosen. Only DLTZ 1-4 problems are considered in this study because the nature of PFs in DTLZ 5-7 is unclear beyond three objectives. The number of decision variables is set to $n = k+m-1$ for DTLZ test instances, where $k = 5$ for DTLZ1 and $k = 10$ for DTLZ2 to DTLZ4. WFG test problems have decision variables $n = k+m-1$, where k is set to 20.

Inverted Generational Distance [22] (IGD) metric is used to calculate the average Euclidean distance between all solutions in the true Pareto front and the non-dominated solutions obtained by the algorithm. A smaller IGD value indicates that the set of non-dominated solutions is closer to the true Pareto front and has a good distribution, IGD value can be calculated as follow:

$$IGD(X, P^*) = \frac{\sum_{x^* \in P^*} d(x^*, X)}{|P^*|} \tag{10}$$

where X denotes the solution set obtained by algorithm, and P^* is the set of points uniformly distributed on the true Pareto front, $d(x^*, X)$ denotes the minimum Euclidean distance from the solution $x^* \in P^*$ to the solution in X. $|P^*|$ is the cardinality of P^*.

The generational distance metric [4] (GD) is used to calculated the average distance between each solution in the set P to the nearest solution in the set P'. The smaller the value is, the better the convergence of the algorithm is. GD value can be calculated as follow:

$$GD(P, P') = \frac{\sqrt{\sum_{y \in P} \min_{x \in P'} dis(x, y)^2}}{|P|} \tag{11}$$

Where P is the solution set obtained by the algorithm, P' is a set of uniformly distributed reference points sampled from the PF, and $dis(x, y)$ denotes the Euclidean distance between the solution y in the solution set P and the solution x in the reference set P'.

4.2 Compared Algorithms and Parameter Settings

In order to evaluate the performance of Two_ArchV, four state-of-the-art algorithms are selected, such as SRA [23], Two_Arch2 [12], NSGA-III [24], and MOEA/D [25].

Population Size: The population size N and the number of weight vectors for different number of objectives are summarized in Table 1. For other peer algorithms such as SRA and Two_Arch2, they use the same parameters as NSGA-III.

Setting up the reproduction operation: In this paper, simulated binary crossover (SBX) and polynomial mutation (PM) are applied. In addition, the crossover probability is $\rho_c = 1.0$ and its distribution index is $\eta_c = 30$. The mutation probability is $p_m = 1/n$, and its distribution index η_m is set to20.

Number of runs and stop condition: Each algorithm's experiments are separately repeated 20 times and terminated after 80,000 function evaluations.

Statistical Test: The Wilcoxon rank sum test [18] with a significance level of 0.05 is used to analyze the results of the experiments, the symbols '+', '=' and '-' indicate that the result of another algorithm

Volume 10 Issue 3, 2024
DOI: 10.6919/ICJE.202403_10(3).0016

is remarkably better, remarkably worse and statistically comparable with proposed by Two_ArchV, respectively.

Parameter settings for each algorithm: For SRA, the indicator parameter k is set to 0.05 and ρ_c is set to a random value in range of [0.4,0.6]. For Two_Arch2, the L_p-norm-based distance p is set $1/m$ (m is the number of objectives) and the parameter $I_{\varepsilon+}$ is set to 0.05. For MOEA/D, the penalty parameter θ of PBI is set to 5, and the neighborhood size is set to $N/10$, where N is the size of population. For other peer algorithms, they do not need to set new parameters.

Table 1. Setting of algorithm population size

Objective number	The weight vectors	NSGA-III	MOEA/D
3	92	92	91
5	210	212	210
8	156	156	156
10	276	276	275
15	136	136	135

5. Experimental Results and Discussion

5.1 Experiment Out

Fig. 3 shows the GD values of CA and DA for two ArchV on DTLZ1 with ten objectives. In this experiment, DTLZ1 problem requires high convergence performance of the algorithm. Therefore, it measures the convergence of CA and DA. From Fig 3, the DA performances better than CA in first 4000 function evaluations. This may be attributed to the boundary protection strategy in CA.In a word, CA obtains better performance in term of convergence than DA. The CA may guide the population to converge toward the true PF, and it does not damage the population diversity.

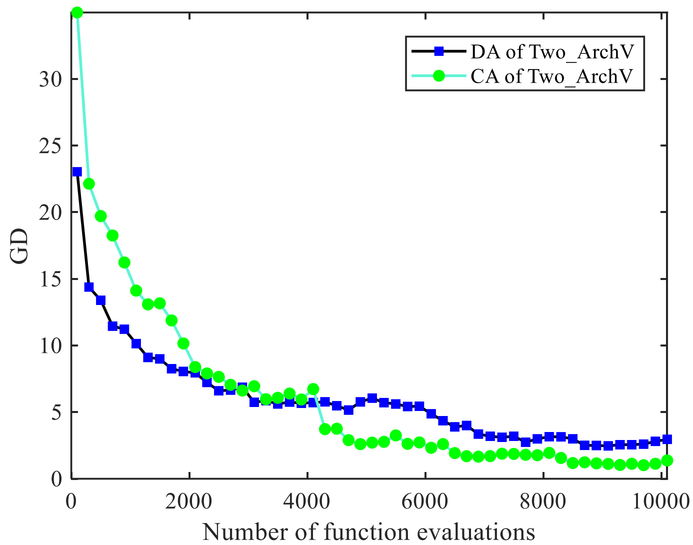

Fig. 3. GD values of the two ArchV's CA and DA on DTLZ1 with ten objectives.

Fig. 4 demonstrates GD values of CA and DA of Two ArchV on DTLZ1 with 10 objectives. In order to analyze the performance of CA and DA in Two_ArchV, it repeats for 30 independent times and terminated by 10,000 function evaluations. Because CA's diversity is limited, Two_Arch2 uses DA as the ultimate result. In this article, DA still outperforms CA in Two_ArchV. Fig 4 makes it evident

Volume 10 Issue 3, 2024
DOI: 10.6919/ICJE.202403_10(3).0016

that during the whole assessment process, DA nearly always has the superior HV value. This indicates that DA would be a better choice to serve as Two_ArchV's ultimate output.

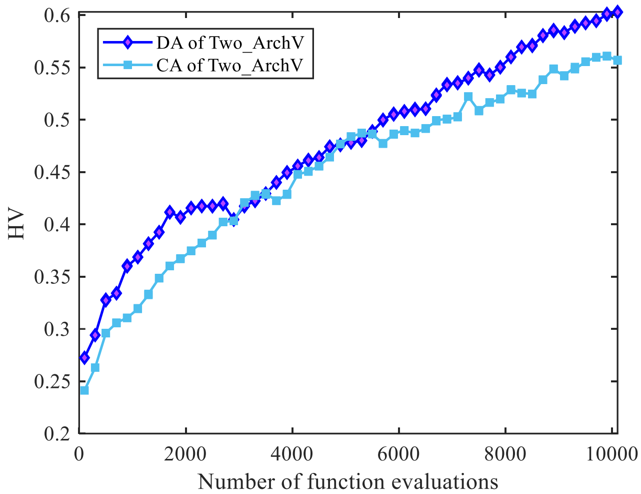

Fig.4. HV values of the two ArchV's CA and DA on WFG8 with ten objectives

5.2 Performance of Algorithm on DTLZ and WFG Problems

Table 2. IGD results obtained by MOEAD, Two_Arch2, NSGA-III, SRA and Two_ArchV on DTLZ1-DTLZ4 on 3,5, 8,10 and 15 objectives, the best and second best mean IGD values of all algorithms for each test instance are highlighted in gray and light gray.

Problem	M	MOEAD	Two_Arch2	NSGA-III	SRA	Two_ArchV
DTLZ1	3	2.8226e+0 (1.24e+1) -	3.4727e-1 (4.32e-1) -	2.6851e-2 (2.81e-2) -	6.5244e-2 (1.55e-1) -	**2.1115e-2 (2.68e-4)**
	5	2.3753e-1 (4.52e-1) -	3.1819e-1 (5.12e-1) -	9.5138e-2 (1.21e-1) -	1.4554e-1 (1.54e-1) -	**6.8343e-2 (4.01e-4)**
	8	2.9766e-1 (2.41e-1) -	1.0350e+0 (4.72e-1) -	1.8259e-1 (2.91e-1) -	2.0180e-1 (2.63e-1) -	**1.1941e-1 (9.52e-4)**
	10	1.6055e-1 (5.82e-2) =	1.3643e+0 (7.48e-1) -	1.9069e-1 (1.15e-1) -	2.1612e-1 (2.15e-1) -	**1.4518e-1 (3.37e-3)**
	15	1.8644e-1 (1.22e-3) +	3.2020e-1 (7.25e-1) -	2.9578e-1 (5.37e-2) =	**1.6641e-1 (8.61e-2) +**	2.8751e-1 (1.07e-1)
DTLZ2	3	5.4648e-2 (2.23e-4) +	6.0402e-2 (1.56e-3) -	**5.4467e-2 (8.15e-6) +**	7.5728e-2 (2.79e-3) -	5.4999e-2 (6.89e-4)
	5	2.1306e-1 (9.63e-4) -	3.2882e-1 (1.94e-1) -	2.1466e-1 (2.55e-3) -	2.2648e-1 (3.52e-3) -	**2.0676e-1 (1.45e-3)**
	8	**3.8520e-1 (4.84e-3) +**	4.3487e-1 (8.25e-3) -	3.9600e-1 (2.74e-2) -	3.8771e-1 (5.03e-3) +	3.8994e-1 (1.90e-3)
	10	5.0676e-1 (1.31e-2) -	5.6370e-1 (1.88e-2) -	5.6914e-1 (4.74e-2) -	**4.7723e-1 (6.65e-3) +**	4.8673e-1 (1.81e-3)
	15	1.0700e+0 (1.10e-1) -	6.4475e-1 (4.84e-2) -	7.3702e-1 (1.50e-2) -	**5.9963e-1 (8.31e-3) +**	6.1264e-1 (2.18e-3)
DTLZ3	3	7.4750e+0 (9.82e+0) -	1.1010e+1 (4.85e+0) -	**5.5363e-2 (1.05e-3) =**	8.7431e-1 (3.55e+0) -	5.5993e-2 (1.24e-3)
	5	1.2391e+1 (1.17e+1) -	1.3761e+1 (8.74e+0) -	5.7670e-1 (1.61e+0) -	3.3263e+0 (6.61e+0) -	**2.3825e-1 (5.70e-2)**
	8	1.1085e+1 (1.01e+1) -	4.9186e+1 (1.25e+1) -	8.0618e+0 (1.42e+1) -	6.0219e+0 (1.04e+1) =	**5.7991e-1 (3.72e-1)**
	10	4.7433e+0 (4.39e+0) -	7.2029e+1 (1.85e+1) -	3.0595e+1 (2.58e+1) -	7.9415e+0 (1.37e+1) -	**1.3096e+0 (9.39e-1)**
	15	1.2636e+0 (7.02e-2) =	1.0552e+1 (8.82e+0) -	1.7450e+0 (1.03e+0) =	**8.3319e-1 (5.73e-1) +**	2.2900e+0 (1.72e+0)
DTLZ4	3	2.9862e-1 (2.49e-1) -	**9.2762e-2 (1.14e-1) +**	1.7216e-1 (2.54e-1) =	1.2191e-1 (1.44e-1) -	9.9486e-2 (1.99e-1)
	5	5.7166e-1 (2.41e-1) -	3.4647e-1 (1.93e-1) -	2.2320e-1 (4.89e-2) -	2.4779e-1 (6.15e-2) -	**2.0840e-1 (9.71e-4)**
	8	7.3460e-1 (1.52e-1) -	4.4340e-1 (7.69e-3) -	4.3877e-1 (7.56e-2) -	4.0738e-1 (2.95e-2) -	**3.8863e-1 (1.25e-3)**
	10	8.0718e-1 (1.08e-1) -	5.2315e-1 (3.31e-2) -	5.7826e-1 (5.27e-2) -	4.9173e-1 (2.27e-2) -	**4.8746e-1 (1.88e-3)**
	15	9.3336e-1 (1.34e-1) -	6.3359e-1 (6.20e-3) -	7.4407e-1 (2.99e-2) -	6.2057e-1 (1.04e-2) -	**6.0992e-1 (1.73e-3)**
+/-/=		3/15/2	1/19/0	1/15/4	5/14/1	/

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

Table 2 presents the average of the IGD results for the comparison algorithms on the DTLZ1-4. Two_ArchV is the most efficient algorithm in terms of the number of best results, and the performance of SRA is very competitive with Two_ArchV. For MOEA/D, it achieves the best performance on DTLZ2, the second best results on 3- objective and 5-objective test instances and the best results on the 8-objective test set. Two_Arch2 performs much worse than its competitors, only performing best on DTLZ4 with 3 objectives. NSGA-III achieves the best results on DTLZ2 and DTLZ3 with 3 objectives. The proposed Two_ArchV algorithm significantly outperforms other compared algorithms on DTLZ2, DTLZ3 and DTLZ4 and shows significant improvements over Two_Arch2 in 19 out of the 20 tested instances.

In order to understand the details of the solution distribution more visually, Fig. 6 plots the final solutions in parallel coordinates for DTLZ3 with 10 objectives.

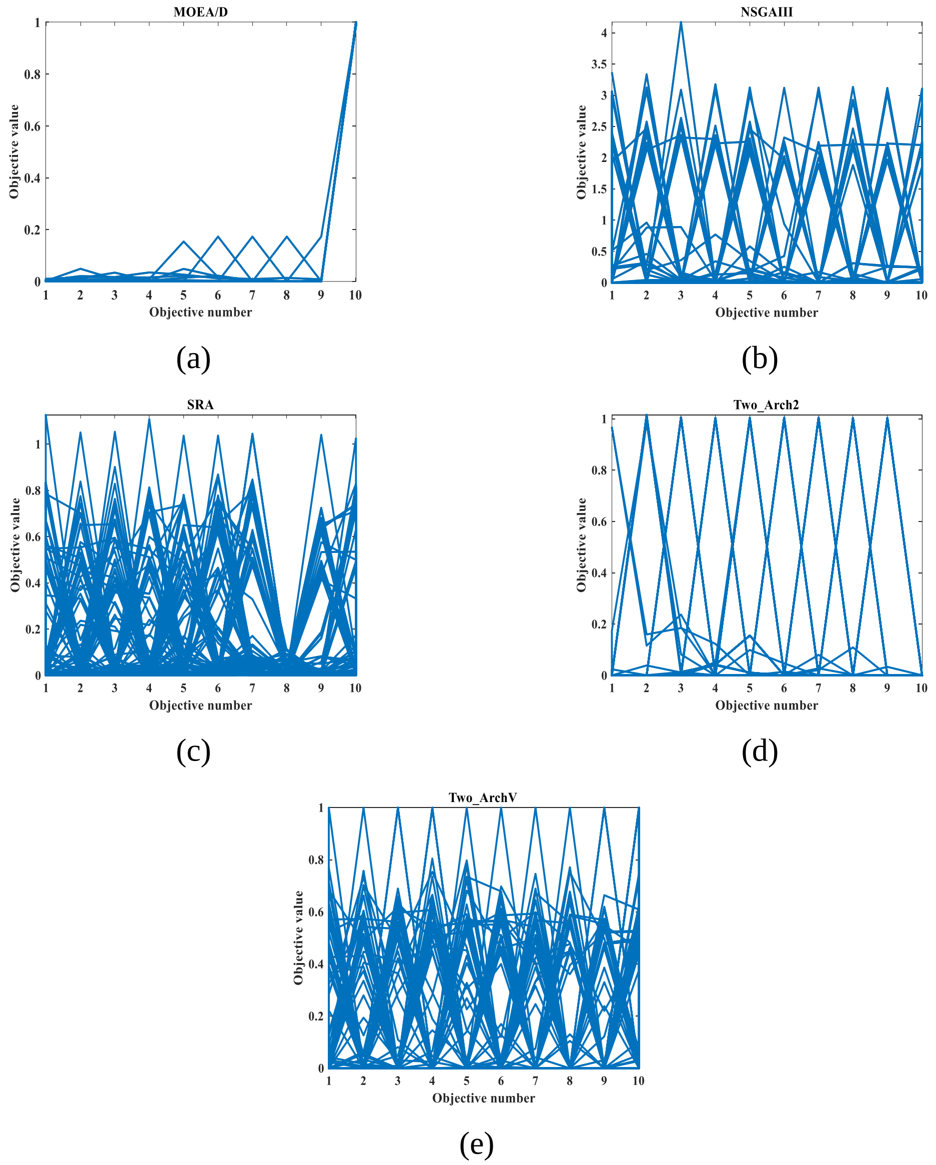

Fig.5. The final solution sets of the five compared algorithms on DTLZ3 problems with 10 objectives by parallel coordinates plot. (a) MOEA/D. (b) NSGA-III. (c) SRA. (d) Two_Arch2. (e) Two_ArchV.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202403_10(3).0016

Table 3. IGD results obtained by MOEAD, Two_Arch2, NSGAIII, SRA and Two_ArchV on WFG1-WFG9 with 3,5, 8,10, and 15 objectives, best and second best mean IGD values of all algorithms for each test instance are highlighted in gray and light gray.

Problem	M	MOEAD	Two_Arch2	NSGAIII	SRA	Two_ArchV
WFG1	3	5.0208e-1 (1.88e-1) -	5.6580e-1 (2.57e-1) -	5.3894e-1 (2.35e-1) -	2.1652e-1 (4.82e-2) -	**1.5419e-1 (4.16e-3)**
	5	1.0943e+0 (1.42e-1) -	1.3269e+0 (1.10e-1) -	1.0882e+0 (3.84e-1) -	6.9288e-1 (3.24e-1) -	**4.6314e-1 (4.70e-3)**
	8	1.7421e+0 (1.06e-1) -	2.0651e+0 (1.23e-1) -	1.0111e+0 (5.54e-2) =	1.3532e+0 (3.99e-1) -	**9.8756e-1 (1.54e-2)**
	10	1.7713e+0 (1.42e-1) -	**1.2100e+0 (6.30e-2) +**	1.5351e+0 (2.39e-1) -	1.3851e+0 (7.91e-2) -	1.2635e+0 (4.31e-2)
	15	2.2991e+0 (4.00e-2) -	**1.7107e+0 (4.88e-2) +**	2.1029e+0 (1.51e-1) -	2.0987e+0 (1.42e-1) -	1.9285e+0 (9.17e-2)
WFG2	3	3.0409e-1 (6.56e-2) -	**1.6375e-1 (7.35e-3) =**	1.6842e-1 (5.08e-3) -	2.2585e-1 (1.55e-2) -	1.6411e-1 (3.02e-3)
	5	9.4779e-1 (6.16e-2) -	5.1337e-1 (1.98e-1) -	5.0601e-1 (8.59e-3) -	5.8267e-1 (3.46e-2) -	**4.9269e-1 (7.30e-3)**
	8	1.8330e+0 (6.57e-2) -	**9.7390e-1 (2.04e-2) +**	1.1317e+0 (1.38e-1) =	1.1735e+0 (7.20e-2) -	1.0721e+0 (2.25e-2)
	10	1.9103e+0 (4.80e-2) -	**1.2393e+0 (5.79e-2) +**	1.4154e+0 (9.70e-2) =	1.3339e+0 (6.76e-2) +	1.3991e+0 (2.16e-2)
	15	2.4535e+0 (3.34e-2) -	**1.7847e+0 (4.47e-2) +**	2.4288e+0 (5.23e-1) -	1.9920e+0 (1.75e-1) -	1.8750e+0 (5.25e-2)
WFG3	3	3.6184e-1 (1.54e-1) -	1.1204e-1 (2.20e-2) -	1.5112e-1 (3.20e-2) -	**6.9237e-2 (5.72e-3) +**	8.7231e-2 (4.49e-3)
	5	1.1931e+0 (3.52e-1) -	**5.1315e-1 (4.12e-2) +**	7.8597e-1 (1.72e-1) -	9.6092e-1 (2.38e-1) -	5.8907e-1 (2.88e-2)
	8	3.9859e+0 (2.15e-1) -	**1.2215e+0 (1.06e-1) +**	2.2282e+0 (3.10e-1) -	2.0325e+0 (7.05e-1) -	1.5089e+0 (1.40e-1)
	10	5.6567e+0 (6.84e-2) -	**1.6759e+0 (2.01e-1) +**	2.9882e+0 (9.31e-1) -	3.4745e+0 (1.07e+0) -	2.3309e+0 (1.94e-1)
	15	8.8252e+0 (1.25e-1) -	**2.7464e+0 (3.47e-1) +**	5.1852e+0 (8.53e-1) -	6.4221e+0 (1.67e+0) -	4.1599e+0 (3.56e-1)
WFG4	3	2.8086e-1 (1.98e-2) -	2.3658e-1 (9.68e-3) -	2.2937e-1 (5.50e-3) -	3.0109e-1 (1.06e-2) -	**2.1743e-1 (3.18e-3)**
	5	2.5197e+0 (2.71e-1) -	1.2143e+0 (1.30e-2) -	1.2281e+0 (3.58e-3) -	1.3249e+0 (2.35e-2) -	**1.1991e+0 (1.17e-2)**
	8	8.4739e+0 (2.43e-1) -	3.4511e+0 (3.14e-2) -	3.5702e+0 (6.72e-2) -	3.5000e+0 (5.04e-2) -	**3.3850e+0 (4.48e-2)**
	10	9.8884e+0 (3.40e-1) -	**5.0324e+0 (1.94e-1) =**	5.9185e+0 (4.40e-2) -	5.4803e+0 (1.63e-1) -	5.0500e+0 (3.66e-2)
	15	1.7612e+1 (3.07e-2) -	9.0803e+0 (5.68e-2) -	1.2129e+1 (2.09e-1) -	9.7525e+0 (4.12e-1) -	**8.7316e+0 (7.14e-2)**
WFG5	3	2.6951e-1 (9.31e-3) -	2.4911e-1 (5.28e-3) -	2.3674e-1 (1.86e-3) -	3.1516e-1 (1.66e-2) -	**2.2794e-1 (2.33e-3)**
	5	2.6206e+0 (1.19e-1) -	1.1995e+0 (1.55e-2) =	1.2107e+0 (1.01e-2) -	1.3213e+0 (2.94e-2) -	**1.1952e+0 (9.40e-3)**
	8	8.1395e+0 (2.75e-1) -	3.4533e+0 (4.64e-2) -	3.5289e+0 (3.57e-3) -	3.4774e+0 (5.47e-2) -	**3.4008e+0 (3.15e-2)**
	10	9.6703e+0 (1.99e-1) -	**5.0585e+0 (1.65e-1) =**	5.8524e+0 (1.83e-2) -	5.4498e+0 (1.90e-1) -	5.0687e+0 (4.26e-2)
	15	1.6629e+1 (1.81e+0) -	9.2322e+0 (1.05e-1) -	1.1705e+1 (2.14e-1) -	1.0082e+1 (9.28e-1) -	**8.6146e+0 (6.95e-2)**
WFG6	3	3.0567e-1 (2.82e-2) -	2.8123e-1 (2.46e-2) -	2.6942e-1 (1.72e-2) -	3.3898e-1 (2.12e-2) -	**2.3290e-1 (9.09e-3)**
	5	3.2341e+0 (2.19e-1) -	1.2674e+0 (2.11e-2) -	1.2382e+0 (1.63e-2) -	1.3745e+0 (2.87e-2) -	**1.2036e+0 (7.39e-3)**
	8	8.5385e+0 (1.29e-1) -	3.5375e+0 (4.63e-2) -	3.5496e+0 (6.44e-3) -	3.5798e+0 (5.87e-2) -	**3.4335e+0 (3.22e-2)**
	10	1.0449e+1 (3.55e-1) -	**5.0680e+0 (1.57e-1) =**	5.9605e+0 (1.39e-1) -	5.4173e+0 (1.49e-1) -	5.0844e+0 (4.84e-2)
	15	1.7514e+1 (4.69e-2) -	8.9009e+0 (5.94e-2) -	1.2081e+1 (1.63e-1) -	1.0007e+1 (7.32e-1) -	**8.5179e+0 (9.40e-2)**
WFG7	3	3.4941e-1 (4.94e-2) -	2.3673e-1 (7.73e-3) -	2.2864e-1 (4.60e-3) -	3.1047e-1 (1.23e-2) -	**2.1918e-1 (2.26e-3)**
	5	2.9003e+0 (2.89e-1) -	1.2063e+0 (1.10e-2) =	1.2417e+0 (1.13e-2) -	1.3765e+0 (3.13e-2) -	**1.2054e+0 (8.29e-3)**
	8	8.6215e+0 (3.96e-1) -	3.4645e+0 (5.93e-2) -	3.5642e+0 (1.22e-2) -	3.4912e+0 (6.88e-2) -	**3.3979e+0 (3.22e-2)**
	10	1.0587e+1 (4.48e-1) -	**5.0235e+0 (1.49e-1) =**	5.9997e+0 (1.01e-1) -	5.4422e+0 (2.18e-1) -	5.0833e+0 (3.75e-2)
	15	1.7294e+1 (1.40e+0) -	8.8042e+0 (8.06e-2) -	1.2251e+1 (2.97e-1) -	9.4993e+0 (2.96e-1) -	**8.6606e+0 (6.21e-2)**
WFG8	3	3.7149e-1 (4.22e-2) -	3.3206e-1 (1.93e-2) -	3.1264e-1 (2.02e-2) -	3.5705e-1 (1.09e-2) -	**2.8508e-1 (3.59e-3)**
	5	2.3254e+0 (3.48e-1) -	1.3738e+0 (1.69e-2) -	1.2819e+0 (2.26e-2) -	1.3623e+0 (3.40e-2) -	**1.2582e+0 (1.16e-2)**
	8	7.2786e+0 (7.32e-1) -	3.9801e+0 (5.30e-2) -	3.8206e+0 (2.69e-1) -	3.7645e+0 (5.61e-1) -	**3.4928e+0 (3.02e-2)**
	10	8.8285e+0 (1.30e+0) -	5.3515e+0 (3.92e-2) -	5.9065e+0 (1.57e-1) -	5.3236e+0 (6.53e-2) -	**5.1992e+0 (3.64e-2)**
	15	1.6699e+1 (2.33e-1) -	9.4670e+0 (1.41e-1) -	1.2079e+1 (2.80e-1) -	9.5556e+0 (1.70e-1) -	**9.0344e+0 (2.14e-1)**
WFG9	3	3.2297e-1 (5.54e-2) -	2.3110e-1 (1.06e-2) -	2.3137e-1 (6.86e-3) -	2.8766e-1 (7.89e-3) -	**2.1817e-1 (1.97e-3)**
	5	2.4974e+0 (1.96e-1) -	1.1932e+0 (1.64e-2) -	1.2061e+0 (1.81e-2) -	1.2732e+0 (3.08e-2) -	**1.1769e+0 (1.22e-2)**
	8	7.8054e+0 (2.84e-1) -	3.5837e+0 (6.40e-2) -	3.5629e+0 (8.83e-2) -	3.4478e+0 (3.36e-2) -	**3.3240e+0 (2.58e-2)**
	10	8747e+0 (6.26e-1) -	5.1947e+0 (1.83e-1) -	5.7905e+0 (6.91e-2) -	5.1768e+0 (9.47e-2) -	**4.9388e+0 (3.44e-2)**
	15	1.5943e+1 (1.69e+0) -	9.1663e+0 (1.36e-1) -	1.1547e+1 (3.93e-1) -	9.2410e+0 (2.82e-1) -	**8.3104e+0 (7.94e-2)**
+/-/=		0/45/0	9/29/7	0/42/3	2/43/0	/

As shown in Fig. 5, the solutions obtained by MOEA/D and Two_Arch2 have a very bad distribution on entire PF. The solutions of the former are unable to spread a wider region for each objective. The later shows bad performance for distribution, and it fails in covering the region between 0.2 and 0.8 of the objective value. As for NSGA-III, a large number of solutions fail in converging into whole PF. Two_ArchV and SRA are able to balance convergence and diversity, there are some differences. The solutions from SRA cannot cover a wider region on 8-th objectives, while Two_ArchV provides a better convergence and distribution than SRA.

Table 3 gives the mean of the results for the comparison algorithms on the WFG test suite in term of IGD metric. The two best algorithms are undoubtedly Two_ArchV and Two_Arch2, and they outperform the other three algorithms on the majority of the test cases. On 30 and 14 of the 45 test problems, respectively, Two_ArchV provides the greatest and second-best IGD scores.In term of Two_Arch2, it produces 14 best results and 15 second best results. As for SRA, it has one best result and seven second best results. NSGA-III has nine second-best results but no best results. The fact that the PFs of WFG test issues are irregular disconnected or mixed, and scaled with varying ranges in each objective may be the reason why MOEA/D based on weight vectors yields the worst performance.A set of well-distributed weight vectors cannot ensure a good distribution of the solutions in population. Unexpectedly, Two_Arch2 performs better than Two_ArchV on WFG3 test problems with most objectives.

WFG3 is featured a degenerated and linear shape. Therefore, some extreme points in Two_ArchV make the population fail to converge to true PF well.

6. Conclusion

This paper proposes a two-archive evolutionary algorithm based on a vector angle approach (Two_ArchV) to handle MaOPs. To address the lack of diversity maintenance in CA, the indicator with boundary protection is used to remove redundant individuals in CA. In order to reduce the loss of Pareto dominance and enhance the convergence of DA, a vector angle-based strategy is assigned to the update of DA. The maximum-vector-angle-first principle and the worse-elimination principle are two components of this strategy. Specially, the maximum-vector-angle-first principle can ensure the excellent distribution of the population by means of vector angles, and the worse-elimination principle can make the poorly converged solutions to be replaced by those good individuals one by one. Two_ArchV is compared with several state-of-the-art MOEAs on WFG and DTLZ benchmarks with up to 15 objectives. The experimental results reveal that Two_ArchV has superior performance compared to other algorithms. In addition, it proves that the incorporation of the two-archive mechanism considerably enhances the convergence and diversity of Two_ArchV for solving MaOPs.

Acknowledgments

This work was supported in part by the National key Research and Development Program of China under Grants (no. 2021YFB3301703), and the Innovation Project of Engineering Research Center of Integration and Application of Digital Learning Technology under Grants (no. 1221028).

References

[1] M. Farina, and Amato P.: On the optimal solution definition for many-criteria optimization problems, 2002 annual meeting of the North American fuzzy information processing society proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622). p. 233-238.

[2] K. Narukawa, and Rodemann T.: Examining the performance of evolutionary many-objective optimization algorithms on a real-world application, 2012 Sixth International Conference on Genetic and Evolutionary Computing. p. 316-319.

[3] R. J. Lygoe, Cary M., and Fleming P. J.: A Real-World Application of a Many-Objective Optimisation Complexity Reduction Process, Evolutionary Multi-Criterion Optimization. p. 641-655.

[4] D. A. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations: Air Force Institute of Technology, 1999.

[5] K. Miettinen, Nonlinear multiobjective optimization: Springer Science & Business Media, 1999.

[6] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction: Springer, 2011.

[7] K. Deb, Pratap A., Agarwal S., et al. : A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on evolutionary computation, Vol. 6 (2002) No. 2, p. 182-197.

[8] Z. Eckart, Marco L., and Lothar T. J. E., Evol. Method Des. Optim. Control Ind. Problem: Improving the strength Pareto evolutionary algorithm for multiobjective optimi-zation, (2001), p. 1-21.

[9] X. Zou, Chen Y., Liu M., et al. : A new evolutionary algorithm for solving many-objective optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 38 (2008) No. 5, p. 1402-1412.

[10] X. Yao: A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm, 2006 International conference on computational intelligence and security. p. 286-291.

[11] B. Li, Li J., Tang K., et al. : An improved two archive algorithm for many-objective optimization, 2014 IEEE Congress on Evolutionary Computation (CEC). p. 2869-2876.

[12] H. Wang, Jiao L., and Yao X.: Two_Arch2: An improved two-archive algorithm for many-objective optimization, IEEE transactions on evolutionary computation, Vol. 19 (2014) No. 4, p. 524-541.

[13] L. Cai, Qu S., and Cheng G.: Two-archive method for aggregation-based many-objective optimization, Information Sciences, Vol. 422 (2018), p. 305-317.

[14] E. Zitzler, and Künzli S.: Indicator-based selection in multiobjective search, PPSN. p. 832-842.

[15] K. Deb, and Agrawal R. B.: Simulated binary crossover for continuous search space, Complex systems, Vol. 9 (1995) No. 2, p. 115-148.

[16] Z. Liang, Luo T., Hu K., et al. : An indicator-based many-objective evolutionary algorithm with boundary protection, IEEE Transactions on Cybernetics, Vol. 51 (2020) No. 9, p. 4553-4566.

[17] X. Zhang, Tian Y., and Jin Y.: A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, Vol. 19 (2014) No. 6, p. 761-776.

[18] J. Derrac, García S., Molina D., et al. : A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation, Vol. 1 (2011) No. 1, p. 3-18.

[19] Y. Xiang, Zhou Y., Li M., et al. : A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, IEEE Transactions on Evolutionary Computation, Vol. 21 (2016) No. 1, p. 131-152.

[20] K. Deb, Thiele L., Laumanns M., et al. , "Scalable test problems for evolutionary multiobjective optimization," Evolutionary multiobjective optimization: theoretical advances and applications, pp. 105-145: Springer, 2005.

[21] S. Huband, Hingston P., Barone L., et al. : A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation, Vol. 10 (2006) No. 5, p. 477-506.

[22] C. A. C. Coello, Evolutionary algorithms for solving multi-objective problems: Springer, 2007.

[23] B. Li, Tang K., Li J., et al. : Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Transactions on Evolutionary Computation, Vol. 20 (2016) No. 6, p. 924-938.

[24] K. Deb, and Jain H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE transactions on evolutionary computation, Vol. 18 (2013) No. 4, p. 577-601.

[25] Q. Zhang, and Li H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on evolutionary computation, Vol. 11 (2007) No. 6, p. 712-731.